Eine "void"-Funktion in C++

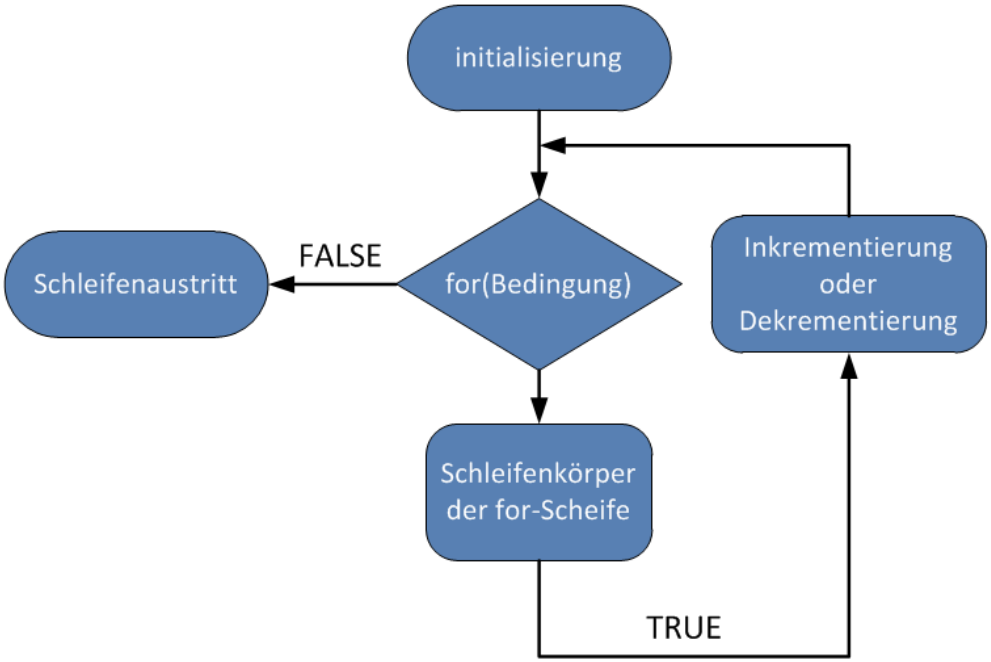
In C++ ist eine "void"-Funktion eine Funktion, die keinen Rückgabewert hat. Das Schlüsselwort "void" wird als Rückgabetyp verwendet, um anzuzeigen, dass die Funktion keine Daten zurückgibt.
Void-Funktionen können verschiedene Aufgaben ausführen, wie zum Beispiel Ausgabe anzeigen, globale Variablen ändern oder mit anderen Teilen des Programms interagieren. Der Hauptzweck besteht darin, Code auszuführen, ohne einen spezifischen Rückgabewert zu liefern.

Faktorielle als Funktion

```
#include <iostream>
using namespace std;


void information(string name, int age, char kurs){
    cout << name << "  " << age <<"  "<< kurs << "
"<<endl;
}

int main() {
information ("Thomas", 19, 'W');

    return 0;
}
```

** Aufbau der for-Schleife

```
for (Initalisierung; Bedingung; Ausdruck)
{
// Anweisungen
// Der Teil zwischen den Klammern wird als Schleifenkörper bezeichnet
}
```



** https://smarduino.com

Rahn Education

main.cpp

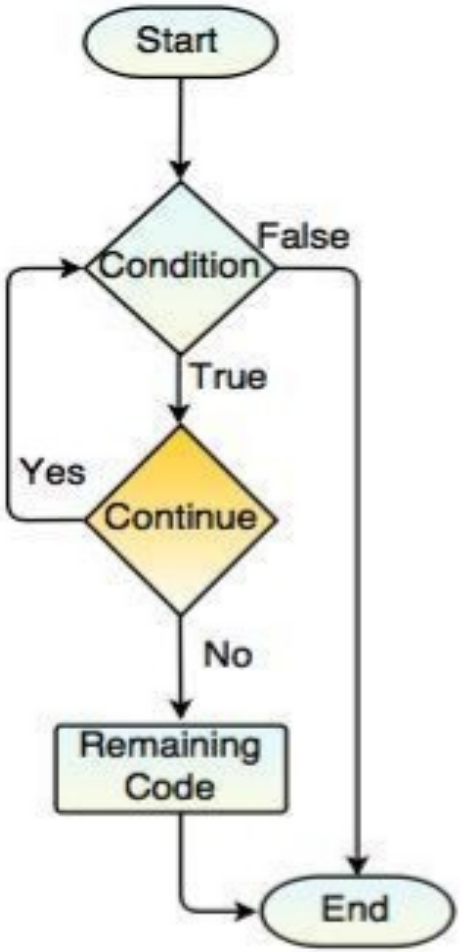
Run

```cpp
// Ein Programm, das die natürlichen Zahlen kleiner oder gleich 10 findet.
#include <iostream>
using namespace std;

int main() {
 for ( int i= 1; i<= 10 ; i= i+1) {
        cout << i << " , " ;
    }
return 0;
}
```

Rahn Education

Continue-Anweisung in C++

```
for (init; condition; update) {
    // code
    if (condition to continue) {
        continue;
    }
    // code
}
```



*www.programiz.com
** www.tutorialride.com

Rahn Education

main.cpp

Run

```cpp
#include <iostream>
using namespace std;
int main() {
for (int i=1; i<= 5; i++){
    cout << i << ", " ;
}
return 0;
}
// Output sind .. 1, 2, 3, 4, 5
```

for loops

Initialize (i)	Bedingung i<=5	Anweisung cout << i ;	Inkrement i++	Runde
1	ja	1	2	I
2	ja	2	3	II
3	ja	3	4	III
4	ja	4	5	IV
5	ja	5	6	V
6	nein	-	-	VI

** Ohne continue oder break Anweisung

Rahn Education

main.cpp

Run

```cpp
#include <iostream>
using namespace std;
int main() {
for (int i=1; i<= 5; i++){
    if (i == 3) continue;
    cout << i << ", " ;
}
return 0;
}
// Output sind .. 1, 2, 4, 5 (ohne 3)
```

for loops

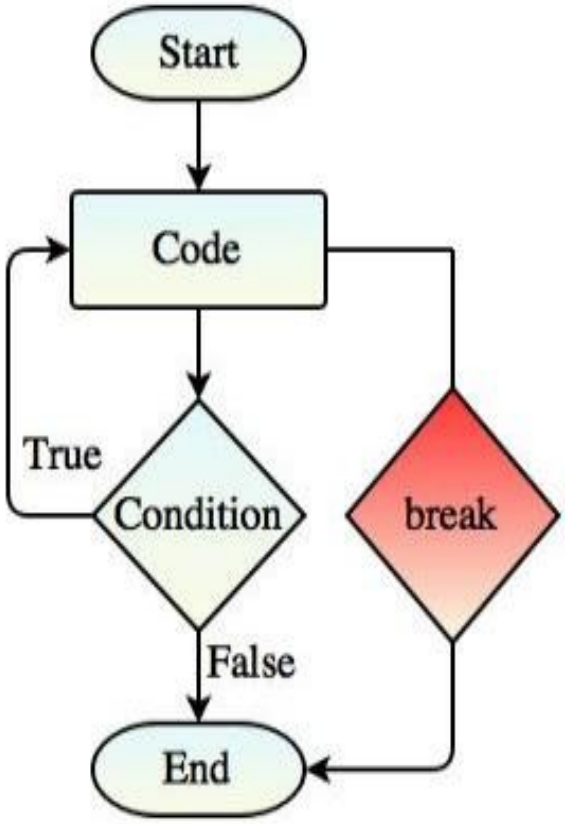
Initialize (i)	Bedingung i<=5 if (i==3) continue;	Anweisung cout << i ;	Inkrement i++	Runde
1	ja, nein	1	2	I
2	ja, nein	2	3	II
3	ja, ja	-	4	III
4	ja, nein	4	5	IV
5	ja, nein	5	6	V
6	nein, nein	-	-	VI

** mit continue

Rahn Education

Break-Anweisung in C++

```
for (init; condition; update) {
    // code
    if (condition to break) {
        break;
    }
    // code
}
```



*www.programiz.com
** www.tutorialride.com

main.cpp

Run

```cpp
#include <iostream>
using namespace std;
int main() {
for (int i=1; i<= 5; i++){
    if (i == 3) break;
    cout << i << ", " ;
}
return 0;
}
// Output sind .. 1, 2 (ohne 3, 4 und 5)
```

for loops

Initialize (i)	Bedingung i<=5 if (i==3) break;	Anweisung cout << i ;	Inkrement i++	Runde
1	ja, nein	1	2	I
2	ja, nein	2	3	II
3	ja, ja	-	4	III
4	ja, ja	-	5	IV
5	ja, ja	-	6	V
6	nein, ja	-	-	VI

** mit break

main.cpp

Run

```cpp
//Finden der geraden Zahlen zwischen a und b

#include <iostream>
using namespace std;
int main() {
int a, b;
cout << " gib mir zwei zahl" << endl;
cin >> a >> b;
int min, max;
if ( a>= b) {
    min = b;
    max = a;
} else {
    min = a;
    max = b;
}
for ( int i = min ; i <= max ; i++) {
 if ( i %2==1) continue;
 cout << i << "  ";
}
    return 0;
}
```

Rahn Education

main.cpp

Run

```cpp
//Finden der Primzahlen  zwischen a und b

#include <iostream>
using namespace std;
int main() {
int a, b;
cout << " gib mir zwei Zahlen" << endl;
cin >> a >> b;
int min, max;
if (a>=b) {
    max=a;
    min=b;
} else {
    max= b;
    min= a;
}
for (int i= min; i <=max; i++) {
    int counter=0;
  for (int k = 1; k<=i;k++) {
      if (i%k==0) counter++;
  }
  if ( counter == 2) cout << i <<" " ;
}
    return 0;
}
```

• While Schleife:

Neben der for-Schleife ist die while-Schleife von großer Bedeutung in C und C++, da diese sich noch vielseitiger als die for-Schleife von einem Programmierer zur Programmsteuerung einsetzen lässt. Wie bei Schleifen üblich, können mit Hilfe einer while-Schleife im Schleifenrumpf enthaltene Anweisungen so oft wiederholt und ausgeführt werden, wie die im Schleifenkopf formulierte Bedingung erfüllt wird.

for Schleife:

```
for (Anfangswert; Prüfe Bedingung; Veränderung des Anfangswertes) {

    Anweisungen
}
```

while Schleife:

```
Anfangswert;

while (Prüfe ob Bedingung erfüllt bzw. auf true oder false) {

    Anweisungen
    Veränderung des Anfangswertes
}
```

- **Beispiel**

main.cpp

Run

Output

```cpp
#include <iostream>
using namespace std;

int main() {
  for (int i=1; i<=10;i++){
      cout << i << "  ";
  }
    return 0;
}
```

```
/tmp/r6nb701F6S.o
1  2  3  4  5  6  7  8  9  10
```

main.cpp

Run

Output

```cpp
#include <iostream>
using namespace std;

int main() {
    int i=1;
  while (i<=10){
      cout << i << "  ";
      i++;
  }
    return 0;
}
```

```
/tmp/r6nb701F6S.o
1  2  3  4  5  6  7  8  9  10
```

main.cpp

Run

```cpp
/*
Ein Programm, um die Kubikzahlen zu finden, die kleiner oder gleich
    1000 ist.
*/

#include <iostream>
#include <cmath>
using namespace std;

int main() {
int x=1;
while (pow(x,3)<=1000) {
    cout << pow(x,3) << "  ";
    x++;
}
    return 0;
}
```

Output

```
/tmp/r6nb7O1F6S.o
1  8  27  64  125  216  343  512  729  1000
```

main.cpp

Run

Output

```cpp
/*
Ein Programm, um die Fakultät mit einer "while"-Schleife zu finden.
*/

#include <iostream>
#include <cmath>
using namespace std;

int main() {
    int n;
    cout << " give me a number"<< endl;
    cin >>n;

int x=1;
int fact=1;
while (x<=n) {
   fact=fact*x;
    x++;
}
cout << n << "! ist " << fact <<endl;
    return 0;
}
```

```
/tmp/r6nb701F6S.o
give me a number
4
4! ist 24
```

do-while-Schleife :

Die "do-while"-Schleife in C++ ist eine Schleifenstruktur, die es ermöglicht, einen bestimmten Codeblock wiederholt auszuführen, solange eine bestimmte Bedingung erfüllt ist.

Der Unterschied zur "while"-Schleife besteht darin, dass die Bedingung am Ende der Schleife überprüft wird, sodass der Codeblock mindestens einmal ausgeführt wird, auch wenn die Bedingung von Anfang an falsch ist.

cpp

```cpp
do {
    // Codeblock, der wiederholt ausgeführt wird
} while (Bedingung);
```

main.cpp

Run

```cpp
/*
Ein Programm, um die Fakultät mit einer "do - while"-Schleife zu finden.
*/

#include <iostream>
#include <cmath>
using namespace std;

int main() {
    int n;
    cout << " give me a number"<< endl;
    cin >>n;

int x=1;
int fact=1;
do {
   fact=fact*x;
    x++;
}
while (x<=n);
cout << n << "! ist " << fact <<endl;
    return 0;
}
```

Output

```
/tmp/r6nb701F6S.o
give me a number
4
4! ist 24
```

"switch-case"-Anweisung :

Die "switch-case"-Anweisung in C++ ermöglicht die Ausführung von unterschiedlichen Codeabschnitten basierend auf dem Wert einer Variablen oder einem Ausdruck. Sie ist eine alternative Methode zum Einsatz mehrerer "if-else"-Anweisungen, wenn verschiedene Bedingungen getestet werden müssen.

- Der Ausdruck wird ausgewertet, und der Programmfluss springt zum passenden "case"-Zweig, dessen Wert mit dem Wert des Ausdrucks übereinstimmt. Wenn kein passender "case"-Zweig gefunden wird, wird der Codeblock innerhalb des "default"-Zweigs ausgeführt.
- **Der optionale "break"-Schlüsselwort wird verwendet, um den Programmfluss zu beenden und aus der "switch-case"-Anweisung herauszuspringen. Wenn kein "break"-Schlüsselwort verwendet wird, führt der Programmfluss direkt in den nächsten "case"-Zweig über und alle folgenden Codeblöcke werden ebenfalls ausgeführt, bis ein "break"-Schlüsselwort erreicht wird.**

```
switch(Ausdruck) {
    case Wert1:
        Anweisungsblock1
        break;
    case Wert2
        Anweisungsblock2
        break;
    default:
        Anweisungsblock3
}
```

main.cpp

Run

Output

```cpp
#include <iostream>
using namespace std;

int main() {
    int vorwahl;
    cout << " vorwahl ist ..."<< endl;
    cin >>vorwahl;
     switch (vorwahl) {
         case 49:
         cout << " Germany" <<endl;
         break;
         case 91:
         cout << " India" <<endl;
         break;
         case 212:
         cout << " Marokko " <<endl;
         break;
         case 970:
         cout << " Palestine " <<endl;
         break;
         case 380:
         cout << " Ukraine " <<endl;
         break;
         default:
         cout << " Ich weiß es nicht..";
         break;
     }
    return 0;
}
```

```
/tmp/r6nb701F6S.o
vorwahl ist ...
49
Germany
```